

COD**E**BABA

JavaScript Blueprint

THIRD EDITION

Mastery official Codebook
ESLAM MOURAD js Path

JS

JavaScript Official Content

CodeBaba.net Guide

The JavaScript Guide shows you how to use [JavaScript](#) and gives an overview of the language. If you need exhaustive information about a language feature, have a look at the [JavaScript reference](#). This Guide is divided into the following chapters.

Chose your learning style

Video (**YouTube Professional Teacher**)

DR. Adel Sabour Video

This visual content helps you implement and simplify the official JavaScript content.

Play list Link : <https://www.youtube.com/playlist?list=PLtrkvrqZCFM7RKWAjNIXy1KTxjQrS992w>

Certifications (**Free & Trusted**)

JavaScript Algorithms and Data Structures

Once you practice and have the fundamentals, you'll apply that knowledge by creating algorithms to manipulate strings, factorialize numbers, and even calculate the orbit of the International Space Station.

Certificate Link : <https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/>

Book (**Recommended reference**)

Eloquent JavaScript

This is a book about JavaScript, programming, and the wonders of the digital. You can read it on-line , free download.

Licensed under a [Creative Commons attribution-noncommercial license](#). All code in this book may also be considered licensed under an [MIT license](#).

Download Link : <https://eloquentjavascript.net/>

Official JavaScript Basis Topics

- [JavaScript basics](#)
- [JavaScript first steps](#)
- [JavaScript building blocks](#)
- [Introducing JavaScript objects](#)

Official JavaScript Topics

Introduction

- [About this guide](#)
- [About JavaScript](#)
- [JavaScript and Java](#)
- [ECMAScript](#)
- [Tools](#)
- [Hello World](#)

Grammar and types

- [Basic syntax & comments](#)
- [Declarations](#)
- [Variable scope](#)
- [Variable hoisting](#)
- [Data structures and types](#)
- [Literals](#)

Control flow and error handling

- [if...else](#)
- [switch](#)
- [try/catch/throw](#)
- [Error objects](#)

Loops and iteration

- [for](#)
- [while](#)
- [do...while](#)
- [continue](#)

- `break`
- `for...in`
- `for...of`

Functions

- Defining functions
- Calling functions
- Function scope
- Closures
- [Arguments](#) & [parameters](#)
- Arrow functions

Expressions and operators

- [Assignment](#) & [Comparisons](#)
- Arithmetic operators
- [Bitwise](#) & [logical operators](#)
- Conditional (ternary) operator

Numbers and dates

- Number literals
- Number object
- Math object
- Date object

Text formatting

- String literals
- String object
- Template literals
- Internationalization
- Regular Expressions

Indexed collections

- Creating an array
- Referring to array elements
- Populating an array
- Understanding length
- Iterating over arrays

- [Array methods](#)
- [Multi-dimensional arrays](#)
- [Using arrays to store other properties](#)
- [Working with array-like objects](#)

Keyed collections

- [Map](#)
- [WeakMap](#)
- [Set](#)
- [WeakSet](#)

Working with objects

- [Objects and properties](#)
- [Creating objects](#)
- [Defining methods](#)
- [Getter and setter](#)

Using classes

- [Declaring a class](#)
- [Various class features](#)
- [Extends and inheritance](#)
- [Why classes?](#)

Promises

- [Guarantees](#)
- [Chaining](#)
- [Error handling](#)
- [Composition](#)
- [Timing](#)

Typed arrays

- [Buffers and views: typed array architecture](#)
- [ArrayBuffer](#)
- [Typed array views](#)
- [DataView](#)
- [Web APIs using typed arrays](#)

- [Examples](#)
 - [Using views with buffers](#)
 - [Multiple views on the same data](#)
 - [Working with complex data structures](#)
 - [Conversion to normal arrays](#)

Iterators and generators

- Iterators
- Iterables
- Generators

Meta programming

- Proxy
- Handlers and traps
- Revocable Proxy
- Reflect

JavaScript modules

- Exporting
- Importing
- Default exports
- Renaming features
- Aggregating modules
- Dynamic module loading

Official JavaScript Advanced Topics

- Client-side JavaScript frameworks
- [Client-side web APIs](#) (Front End)
- Language overview
- JavaScript data structures
- Equality comparisons and sameness
- Enumerability and ownership of properties
- Closures
- Inheritance and the prototype chain
- Memory Management
- Concurrency model and Event Loop